

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-162013

(43)Date of publication of application : 19.06.1998

(51)Int.Cl.

G06F 17/30

(21)Application number : 08-317918

(71)Applicant : NIPPON TELEGR & TELEPH CORP <NTT>

(22)Date of filing : 28.11.1996

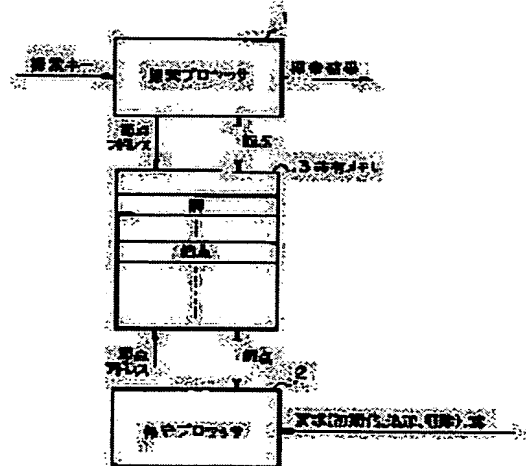
(72)Inventor : TAKAHASHI NAOHISA
MARUYAMA MITSURU
SANEI TAKESHI
OGURA TAKESHI
KAWANO TETSUO
YAGI SATORU

(54) DIGITAL SEARCHING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a digital searching device which executes a searching processing on a digital search tree at a high speed even if the number of headers in the digital search tree is increased, even if the request frequency of the searching processing, an elimination processing and an addition processing increase or even if the requests of the searching processing are continuously outputted.

SOLUTION: In the digital searching device, the digital search tree is kept in a common memory 3 and a maintenance processor 2 executes the initialization processing of the digital search tree, the addition processing of leaves and nodes and the elimination processing of the leaves and the nodes. A searching processor 1 executes the searching processing of the digital search tree. The maintenance processor 2 and the searching processor 1 independently operate and they access to the common memory 3 so as to operate the digital search tree in parallel.



LEGAL STATUS

[Date of request for examination] 21.12.1998

[Date of sending the examiner's decision of rejection] 21.11.2001

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3284064

[Date of registration] 01.03.2002

[Number of appeal against examiner's decision of rejection] 2001-22980

[Date of requesting appeal against examiner's decision of rejection] 21.12.2001

[Date of extinction of right]

BEST AVAILABLE COPY

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-73329

(43)公開日 平成11年(1999) 3月16日

(51)Int.Cl.⁸

識別記号

F I

G 0 6 F 9/45

G 0 6 F 9/44

3 2 2 H

12/08

12/08

S

審査請求 未請求 請求項の数6 OL (全 23 頁)

(21)出願番号 特願平10-13878

(22)出願日 平成10年(1998) 1月27日

(31)優先権主張番号 特願平9-166777

(32)優先日 平 9 (1997) 6月24日

(33)優先権主張国 日本 (J P)

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 津幡 真太郎

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

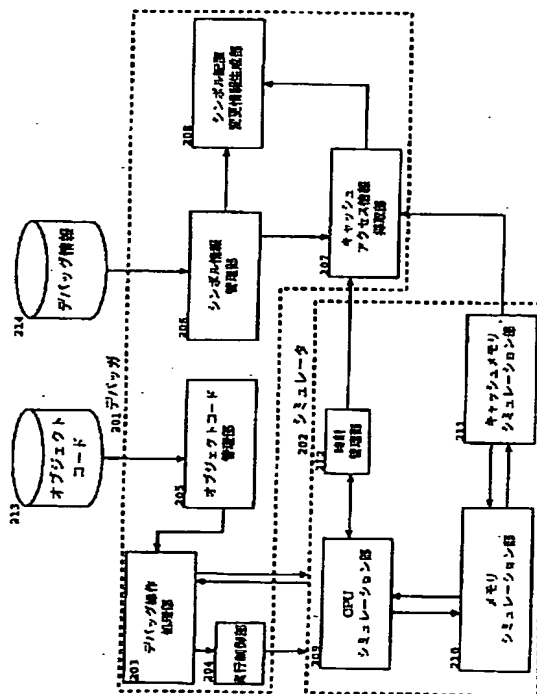
(74)代理人 弁理士 宮井 暎夫

(54)【発明の名称】 ソフトウェア開発支援システム

(57)【要約】

【課題】 キャッシュメモリへのアクセスの状況を高級言語レベルで検出し、キャッシュメモリのヒット率が向上するようにシンボルの割付情報を生成することにより、キャッシュメモリ搭載プロセッサ上のプログラム開発を支援するソフトウェア開発支援システムを提供する。

【解決手段】 プロセッサの動作をシミュレーションするシミュレータ202と、キャッシュメモリへのアクセスに関する情報を採取するキャッシュアクセス情報採取部207と、プログラムのシンボル情報を管理するシンボル情報管理部206と、プログラムのシンボルの配置を変更するためのシンボル配置変更情報を生成するシンボル配置変更情報生成部208とを備えている。



1

【特許請求の範囲】

【請求項1】 キャッシュメモリを搭載するプロセッサ上で実行されるプログラムのソフトウェア開発支援システムであって、

前記プロセッサの動作をシミュレーションするシミュレーション手段と、このシミュレーション手段から前記キャッシュメモリへのアクセスに関する情報を採取するキャッシュアクセス情報採取手段と、前記プログラムのシンボル情報を管理するシンボル情報管理手段と、前記プログラムのシンボルの配置を変更するためのシンボル配置変更情報を生成するシンボル配置変更情報生成手段とを備えたソフトウェア開発支援システム。

【請求項2】 キャッシュアクセス情報採取手段は、キャッシュメモリでのエントリの競合性ミスに関する情報を採取する請求項1記載のソフトウェア開発支援システム。

【請求項3】 シンボル配置変更情報生成手段は、キャッシュメモリでのエントリの競合性ミスを低減するための情報を生成する請求項1または請求項2記載のソフトウェア開発支援システム。

【請求項4】 キャッシュメモリを搭載するプロセッサ上で実行されるプログラムのソフトウェア開発支援システムであって、

前記プロセッサの動作をシミュレーションするシミュレーション手段と、このシミュレーション手段から前記キャッシュメモリへのアクセスに関する情報を採取するキャッシュアクセス情報採取手段と、前記プログラムのシンボル情報を管理するシンボル情報管理手段と、前記シンボル情報を用いて前記キャッシュアクセスの情報を表示する表示手段とを備えたソフトウェア開発支援システム。

【請求項5】 キャッシュアクセス情報採取手段は、アクセス対象のアドレスおよびアクセス時のプログラムカウンタ値、スタックポインタ値ならびにアクセスがヒットかどうかの情報を採取する請求項4記載のソフトウェア開発支援システム。

【請求項6】 表示手段は、キャッシュメモリでのエントリ別に分類して表示する請求項4または請求項5記載のソフトウェア開発支援システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、キャッシュメモリを搭載するプロセッサ上で実行されるプログラムの開発を支援するソフトウェア開発支援システムに関するものである。

【0002】

【従来の技術】 近年の電子技術の発展により、家電機器などの組み込み機器においてもマイクロコンピュータなどプロセッサを組み込み、プログラム開発を伴った機器開発が行われている。加えて、組み込み機器の高機能化

2

に伴い、プログラムが大規模になり、プログラムの開発および保守の効率化のため、C言語など高級言語によるプログラム開発が行われている。

【0003】 また、プログラムの大規模化に従い、大容量のメモリがプログラムのコード格納やデータ格納のために必要とされている。さらに、プログラムの処理の高速化のため、メモリアccessを高速にするキャッシュメモリ搭載のプロセッサが組み込み機器に用いられている。図3に、キャッシュメモリ搭載のプロセッサのシステム構成の一例を示す。図3において、点線で囲まれた部分がプロセッサであり、601はCPU（中央演算装置）であり各種演算を行う。602は、プロセッサの外部に存在する主記憶装置であり、プログラム実行コード（オブジェクトコード）およびデータを記憶する。603はキャッシュメモリであり、主記憶装置へのアクセスを高速化するために機能する。604はバスであり、このバスを用いて、CPU601、主記憶装置602、キャッシュメモリ603の間でデータが転送される。

【0004】 このようなプロセッサにおける、キャッシュメモリの機能について説明する。主記憶装置602上のアドレスに対する読み込みアクセスの場合には、そのアドレスおよびそのデータがキャッシュメモリ603の対応するアドレス上に登録されていれば（キャッシュヒットと呼ぶ）、そのキャッシュメモリ603上のデータが参照され、そのアクセスが高速に実行される。

【0005】 また、アドレスおよびデータがキャッシュメモリ603上に未登録であれば（キャッシュミス）、主記憶装置602上のデータが読み込まれると同時に、アドレスとデータがキャッシュメモリ603上に登録される。この場合のアクセス時間は、はキャッシュヒット時に比べて大きい。同様に、主記憶装置602上のアドレスに対する書き込みアクセスの場合は、そのアドレスおよびそのデータがキャッシュメモリ603の対応するアドレス上に登録されていれば（キャッシュヒット）、そのキャッシュメモリ603上のデータのみが更新されることにより、そのアクセスが高速に実行される。

【0006】 また、そのアドレスおよびデータがキャッシュメモリ603上に未登録であれば（キャッシュミス）、主記憶装置602上のデータおよびそのアドレスとデータがキャッシュメモリ603上に登録されると同時にキャッシュメモリ603上のデータのみが更新される。この場合のアクセス時間は、キャッシュヒット時に比べて大きい。

【0007】 なお、主記憶装置上のアドレスとキャッシュメモリ上への対応づけの方式には、主記憶装置上の1つのアドレスに1つのキャッシュメモリ上のアドレスが対応づくダイレクトマップ方式や、2つのキャッシュメモリ上のアドレスが対応づく2ウェイセットアソシエティブ方式など幾つかの方式がある。また、キャッシュメモリの方式によってキャッシュメモリへの書き込みと

10

20

30

40

50

3

同時につねに主記憶装置上へも書きこみが行われる方式（ライトスルー方式）もある。

【0008】以上説明したようなキャッシュメモリ603を搭載したプロセッサにおいて、実行されるプログラムの性能評価および性能向上は、プログラム実行時のキャッシュヒット、キャッシュミス（以下「ミス」）の状況を考慮し行う必要がある。加えて、組み込み機器開発においては、リアルタイム応用など高速処理実現と低コスト実現の両立が求められ、より正確な性能評価およびそれに基いた性能向上が求められる。

【0009】しかし、従来は、性能評価として、実機での実行時間の測定や、またシミュレーションによるサイクル数の見積もりなどが行われてきた。また、キャッシュ搭載のプロセッサを対象とする場合は、性能評価の指標としてキャッシュのヒット率つまりアクセス回数に対するキャッシュヒットの比率が専ら用いられてきた。

【0010】このような性能評価を行い、速度要求を満たしていないプログラムのサブルーチンなど特定部分については、利用者がプログラムを変更し、速度の改善を図ってきた。プログラム変更の一つの方法は、キャッシュヒット率を向上させ、メモリアccessの速度を向上させることであり、例えば、キャッシュヒット率が向上するように、プログラムで使用するデータの割付けアドレスを変更することである。

【0011】

【発明が解決しようとする課題】しかしながら、従来の方法においてキャッシュヒット率だけではその詳細な状況は判らず、また、シミュレーションにおけるプログラム動作のトレース結果では、トレースされる対象がアドレスだけであり、実際に利用者が変更する対象である高級言語レベルのプログラムには対応しておらず、性能向上のためのプログラム変更が困難であるという問題を有していた。

【0012】この発明は、上記の課題に鑑み、キャッシュメモリへのアクセスの状況を高級言語レベルで検出し、キャッシュメモリのヒット率が向上するようにシンボルの割付情報を生成することにより、キャッシュメモリ搭載プロセッサ上のプログラム開発を支援するソフトウェア開発支援システムを提供することを目的とする。また、キャッシュミスの分類の1つとして、キャッシュメモリの1つのエントリに複数の主記憶上のブロック（アドレス）が対応するために起因する競合性ミスがある。このキャッシュミスは、複数の主記憶上のブロックが1つのエントリに対応づかないようにプログラムを変更してやれば解決できる可能性があるキャッシュミスである。

【0013】この発明は、したがって上記の課題に鑑み、キャッシュミスのうち特に競合性ミスを検出し競合しているシンボルを競合しないようにシンボルの割付情報を生成することにより、キャッシュメモリ搭載プロセ

4

ッサ上のプログラム開発を支援するソフトウェア開発支援システムを提供することを目的とする。さらにこの発明は、上記の課題に鑑み、キャッシュメモリへのアクセスの状況を高級言語レベルで利用者に表示し、キャッシュメモリ搭載プロセッサ上のプログラムを支援するソフトウェア開発支援システムを提供することを目的とする。

【0014】

【課題を解決するための手段】請求項1記載のソフトウェア開発支援システムは、キャッシュメモリを搭載するプロセッサ上で実行されるプログラムのソフトウェア開発支援システムであって、プロセッサの動作をシミュレーションするシミュレーション手段と、このシミュレーション手段からキャッシュメモリへのアクセスに関する情報を採取するキャッシュアクセス情報採取手段と、プログラムのシンボル情報を管理するシンボル情報管理手段と、プログラムのシンボルの配置を変更するためのシンボル配置変更情報を生成するシンボル配置変更情報生成手段とを備えたものである。

【0015】請求項1記載のソフトウェア開発支援システムによれば、キャッシュアクセス情報採取手段で採取された情報と、シンボル情報管理手段で管理されているシンボル情報から、プログラムのメモリアccess時の速度が高速になるようにシンボルの配置を変更するためのシンボルの配置変更情報を生成することができる。これにより、キャッシュメモリへのアクセスの状況が対象プログラムが記述される高級言語レベルで検知可能となり、アクセス速度を低下させるシンボルの配置を変更し、プログラムの実行性能を向上させることが可能となる。

【0016】請求項2記載のソフトウェア支援開発システムは、請求項1において、キャッシュアクセス情報採取手段が、キャッシュメモリでのエントリの競合性ミスに関する情報を採取するものである。請求項2記載のソフトウェア支援開発システムによれば、請求項1と同様な効果のほか、キャッシュアクセス情報採取手段で採取された情報からキャッシュ競合性ミスを検出し、シンボル情報と関連づけて、シンボルの配置変更情報をシンボル配置変更情報生成手段で生成することができる。これにより、キャッシュメモリ搭載のプログラムを開発する際に、キャッシュメモリへのアクセスの状況特に競合性ミスをシンボルレベルないし高級言語レベルで検出し、キャッシュメモリで競合性ミスが発生するシンボルの配置を変更する情報を生成するなど、その競合を解消するためにシンボルの配置を変更することが容易にでき、プログラム開発の効率化が図れる。

【0017】請求項3記載のソフトウェア支援開発システムは、請求項1または請求項2において、シンボル配置変更情報生成手段が、キャッシュメモリでのエントリの競合性ミスを低減するための情報を生成するものであ

5

る。請求項3記載のソフトウェア支援開発システムによれば、請求項1または請求項2と同様な効果のほか、競合性のキャッシュミスが発生したシンボルをキャッシュされていない空のエントリに対応するアドレスに変更するための情報を生成するなど、キャッシュメモリの競合性ミスを低減させるようなシンボルの配置変更情報が生成でき、より一層のプログラムの性能向上を図れる。

【0018】請求項4記載のソフトウェア支援開発システムは、キャッシュメモリを搭載するプロセッサ上で実行されるプログラムのソフトウェア支援開発システムであって、プロセッサの動作をシミュレーションするシミュレーション手段と、このシミュレーション手段からキャッシュメモリへのアクセスに関する情報を採取するキャッシュアクセス情報採取手段と、プログラムのシンボル情報を管理するシンボル情報管理手段と、シンボル情報を用いてキャッシュアクセスの情報を表示する表示手段とを備えたものである。

【0019】請求項4記載のソフトウェア支援開発システムによれば、キャッシュアクセス情報採取手段で採取された情報と、シンボル情報管理手段で管理されているシンボル情報を表示手段により関連付けて表示することができるので、キャッシュメモリへのアクセスの状況が、対象プログラムが記述される高級言語レベルで表示可能となり、評価性能に基いたプログラムの変更が容易になる。したがって、キャッシュメモリ搭載のプログラムを開発し評価する際に、キャッシュメモリへのアクセス状況を高級言語レベルで知ることができ、プログラム開発の効率化が図れる。

【0020】請求項5記載のソフトウェア支援開発システムは、請求項4において、キャッシュアクセス情報採取手段が、アクセス対象のアドレスおよびアクセス時のプログラムカウンタ値、スタックポインタ値ならびにアクセスがヒットかどうかの情報を採取するものである。請求項5記載のソフトウェア支援開発システムによれば、請求項4と同様な効果がある。

【0021】請求項6記載のソフトウェア支援開発システムは、請求項4または請求項5において、表示手段が、キャッシュメモリでのエントリ別に分類して表示するものである。請求項6記載のソフトウェア支援開発システムによれば、請求項4または請求項5と同様な効果がある。

【0022】

【発明の実施の形態】以下、この発明の実施の形態について説明する。この発明の第1の実施の形態を図1から図16により説明する。図1は、この発明の第1の実施の形態におけるソフトウェア支援開発システムに適用されるシステムの一例であり、高級言語によるソフトウェア開発およびそのシミュレーション、ならびにデバッグに係る全体的なシステムである。

【0023】図1において、105は、開発の対象とな

6

るアプリケーションを高級言語で記述したソースプログラムである。101はコンパイラであり、ソースプログラム105を入力し、対象とするプロセッサに対応したオブジェクトコード106と、デバッグ情報107を出力する。

【0024】102は、ソフトウェア開発支援システムであり、デバッグ103と、対象プロセッサの動作をシミュレーションするシミュレーション手段であるシミュレータ104とから構成される。開発支援システム102は、オブジェクトコード106とデバッグ情報107を入力し、アプリケーションのシミュレーションおよび高級言語レベルでのデバッグを実現する。

【0025】図9に、開発支援システム102が対象とするプロセッサの構成図を示す。図9は図3と比べて、キャッシュメモリが命令キャッシュメモリ1303とデータキャッシュメモリ1304に別れている点が異なる。よって、命令キャッシュメモリ1303とデータキャッシュメモリ1304についてのみ説明する。命令キャッシュメモリ1303は、命令の読み込み動作（フェッチ動作）時に、用いられるキャッシュメモリであり、データキャッシュメモリ1304は、データの読み込み動作（データリード動作）時とデータの書きこみ動作（データライト動作）時に用いられるキャッシュメモリである。

【0026】また、命令キャッシュメモリ1303は、内部にタグアレイ1306およびデータアレイ1307を備え、同様にデータキャッシュメモリ1304は、タグアレイ1308とデータアレイ1309を備える。タグアレイ1306、1308は、主記憶装置1302上のアドレスとキャッシュメモリのデータアレイ1307、1309上のアドレスを対応付けるタグ情報を保持する。また、データアレイ1307、1309は、上記のように対応付けられたデータの内容自体を保持する。

【0027】図4は、プロセッサのCPUの構成とともに、メモリとの関係を示す。図4において、701はCPUであり、プログラム実行時に主にデータを格納するデータレジスタ（D0、D1、D2、D3）703、704、705、706と、主にアドレスを格納するアドレスレジスタ（A0、A1、A2、A3）707、708、709、710と、演算結果フラグを格納するプロセッサ状態語レジスタ（PSW）711と、プログラム実行アドレスを指し示すプログラムカウンタレジスタ（PC）と、スタック領域のアドレスを指し示すスタックポインタレジスタ（SP）と、各種演算を行う演算器714から構成される。

【0028】702は、メモリ領域の概念図である。702に示すように、メモリ領域は、プログラム自体である実行コードを格納する実行コード格納領域715と、主にグローバルなデータを格納するデータ領域716と、主にローカルなデータを格納するスタック領域71

7から構成される。このメモリ領域には、すべてアドレスが割振られ、CPU701からはアドレスによりアクセスが行われる。この構成において、プロセッサは、実行コード格納領域715に格納されている命令を読み込み、読み込んだ命令に従って各種レジスタあるいはメモリ上のデータを用いて演算を行う。

【0029】このようなプロセッサでは、プロセッサ(CPU)とメモリ(主記憶)の間のデータの転送は、(1)メモリからプロセッサへの命令の読み込み(フェッチ)、(2)メモリからプロセッサへのデータの読み込み(データリード)、(3)プロセッサからメモリへのデータの書きこみ(データライト)の3種類に区分される。

【0030】表1に、ソースプログラムの一例を示す。

【0031】

【表1】

```
01:  int x, y;
02:
03:  void main() {
04:      int i;
05:      x = y = 0;
06:      for (i=0; i<100; i++) {
07:          x = i*i - i;
08:          y = y + x*2;
09:      }
10: }
```

```
01:  _TEXT  SECTION
02:
03:  0x100      add    -4, SP
04:  0x104      clr    D0
05:  0x108      mov    D0, (0x410)
06:  0x10C      mov    D0, (0x400)
07:  0x110      mov    D0, (SP)
08:  0x114      jmp    0x150
09:
10:  0x118      mov    (SP), D0
11:  0x11C      mov    D0, D1
12:  0x120      mul    D1, D0
13:  0x124      mov    (SP), D1
14:  0x128      sub    D1, D0
15:  0x12C      mov    D0, (0x400)
16:  0x130      mov    D0, D1
17:  0x134      mov    (0x410), D0
18:  0x138      add    D1, D0
19:  0x13C      add    D1, D0
20:  0x140      mov    D0, (0x410)
21:  0x144      mov    (SP), D0
22:  0x148      inc    D0
23:  0x14C      mov    D0, (SP)
24:
25:  0x150      mov    (SP), D0
26:  0x154      cmp    100, D0
27:  0x158      bge    0x160
28:  0x15C      jmp    0x118
29:
30:  0x160      ret
31:
32:
33:  _DATA  SECTION
34:  0x400      DS      4
35:  0x410      DS      4
```

【0035】表2では、説明のため機械語命令と対応するアセンブラ記述を示しているが、本質的には前記の2つは同一の情報を表現する。また、全ての機械語命令は、4バイトの長さのビットパターンである。なお、表2において、各行の左端の『数字:』は、説明のために

*【0032】表1に示すソースプログラムは、高級言語の一つであるC言語によって記述されている。なお、表1において、各行の左端の『数字:』は、説明のために付与した行番号である。表1のソースプログラムにおいて、1行目は、グローバル変数でint(整数)型変数xとyの宣言を示す。3行目から10行目までが、関数mainの定義部分である。関数mainでは、4行目に示すように、ローカル変数でint(整数)型変数iを用いて、6行目から9行目に示すように、変数iを0から99まで増加させて、7、8行目に示す演算を繰り返し行う処理が記述されている。

【0033】表2に、オブジェクトコードに対応するアセンブラ記述による説明図を示す。オブジェクトコードは、対象とするプロセッサの機械語命令で表現され、アドレスとそのアドレスに格納する値が情報として記述される。

【0034】

【表2】

付与した行番号であり、左端2番目の16進数による表記は、その命令あるいはデータが格納されるアドレスを示す。例えば、3行目の命令『add -4, SP』に対応するオブジェクトコードは、アドレス0x100番地に格納され、33行目に示すデータxに対応するデー

タは、アドレス0x400番地に格納される。

【0036】表2において、1行目から30行目までは、実行コード部分である。1行目は、アセンブラ記述での擬似命令であり、1行目に続く部分が実行コードであることを示す。また、33行目から35行目は、データ領域部分であり、33行目は、以下に続く部分がデータ領域であることを示すアセンブラ擬似命令である。34行目は、アドレス0x400番地以降に4バイト分の領域を割当ることを示す。同様に、35行目は、アドレス0x410番地以降に4バイト分の領域を割当ることを示す。

【0037】後述するシンボル情報によって、アドレス0x100番地がC言語プログラム上の関数mainの先頭番地に対応付けられ、また、アドレス0x400番地が変数xに、同様にアドレス0x410番地が変数yに対応付けられる。下記に3行目から30行目に記述されている実行コードで用いられているアセンブラ記述とプロセッサの命令の対応を示す。

【0038】(1) add 『加算データ』, 『被加算データ』

『加算データ』で示されるレジスタあるいはメモリの内容あるいは即値を『被加算データ』で示されるレジスタの内容に加算して、結果を『被加算データ』で示されるレジスタに格納する。例えば、3行目の記述は、即値-4をスタックポインタレジスタ(SP)の内容に加算して、加算結果をスタックポインタレジスタに格納する命令である。また、18行目の記述は、データレジスタD0の内容をデータレジスタD1の内容に加算して、加算結果をデータレジスタD1へ格納する命令である。

【0039】(2) clr 『対象レジスタ』

『対象レジスタ』で示されるレジスタの内容を値0にする。例えば、4行目の記述は、データレジスタD0の内容を0にする命令である。

(3) inc 『対象レジスタ』

『対象レジスタ』で示されるレジスタの内容を1増加して格納する。例えば、22行目の記述は、データレジスタD0の内容を1増加して格納する命令である。

【0040】(3) sub 『減算データ』, 『被減算データ』

『減算データ』で示されるレジスタあるいはメモリの内容あるいは即値を『被減算データ』で示されるレジスタの内容から減算して、結果を『被減算データ』で示されるレジスタに格納する。例えば、14行目の記述は、データレジスタD1の内容をデータレジスタD0の内容から減算して、減算結果をデータレジスタD0に格納する命令である。

【0041】(4) cmp 『減算データ』, 『被減算データ』

『減算データ』で示されるレジスタあるいはメモリの内容あるいは即値を『被減算データ』で示されるレジスタ

の内容から減算して、結果のフラグ(結果が正である、結果が負である、結果が0である)をプロセッサ状態語レジスタPSWに反映する。sub命令と異なり、減算結果は、レジスタには格納されない。例えば、26行目の記述は、データレジスタD0から即値100を減算して、その結果フラグをプロセッサ状態語レジスタPSWに反映する命令である。

【0042】(5) mov 『転送元データ』, 『転送先データ』

『転送元データ』で示されるレジスタあるいはメモリの内容あるいは即値を『転送先データ』で示されるレジスタあるいはメモリへ転送する。例えば、5行目の記述は、データレジスタD0の内容をアドレス0x410番地から始まる1ワード(4バイト)分のメモリ上の領域に転送する命令である。また、例えば、10行目の記述は、スタックポインタレジスタSPの内容が指し示すメモリ上の1ワード分の領域の内容を、データレジスタD0へ格納する命令である。

【0043】(6) jmp 『ジャンプ先』

『ジャンプ先』で示されるアドレスへ、プログラム実行の制御を変更する。例えば、8行目の記述は、0x150番地(25行目)にプログラム実行の制御を変更する命令である。なお、このようなジャンプ命令以外は、プログラム実行は逐次的に行われる。例えば、0x100番地(3行目)の命令の次は0x104番地(4行目)の命令が実行される。

【0044】(7) mul 『乗算データ』, 『被乗算データ』

『乗算データ』で示されるレジスタの内容を、『被乗算データ』で示されるレジスタの内容に乗じて、乗算結果を『被乗算データ』で示されるレジスタに格納する。例えば、12行目の記述は、データレジスタD1の内容をデータレジスタD0の内容に乗算して、その乗算結果をデータレジスタD0に格納する命令である。

【0045】(8) bge 『ジャンプ先』

プロセッサ状態語レジスタのフラグが正あるいは0の場合に、『ジャンプ先』で示されるアドレスへ、プログラム実行の制御を変更する。例えば、27行目の記述は、フラグが正あるいは0の場合に、0x160番地(30行目)にプログラム実行の制御を変更する命令である。

【0046】(9) ret

サブルーチン呼び出し元へプログラムの制御を変更する。しかし、オブジェクトコードには、前記に説明したようなC言語での関数あるいは変数との対応付けの情報はなく、実際には次に説明するシンボル情報で示される。表3に、デバッグ情報の一例であるシンボル情報を示す。

【0047】

【表3】

11

シンボル情報

```

01: 関数 main
02: 開始アドレス 0x100
03: 引数 void
04: 返り値 void
05: ローカル変数
06: int i (SP)
07:
08:
09: 変数 x
10: 開始アドレス 0x400
11: int型
12:
13: 変数 y
14: 開始アドレス 0x410
15: int型

```

【0048】なお、表3において、各行の左端の『数字:』は、説明のために付与した行番号である。シンボル情報では、高級言語上の変数名や関数名などシンボルとオブジェクトコード上でのアドレスとの対応付けを行う情報である。表3において、1行目から6行目では、シンボル“main”が関数であり、その開始アドレスが0x100番地であり、関数の引数および戻り値がvoid型であり、加えて関数main内のローカル変数iがスタックポインタ(SP)レジスタが指すアドレスに割付けられており、その型がint(整数)型であることを示している。

【0049】また、9行目から11行目までには、シンボル“x”がグローバル変数であり、その開始アドレスが0x400番地であり、その型がint(整数)型であることを示している。同様に、13行目から15行目では、シンボル“y”がグローバル変数であり、その開始アドレスが0x410番地であり、その型がint(整数)型であることを示している。

【0050】図2に、前記開発支援システム102の構成の一例を示す。図2において、201はデバッグ103であり、デバッグ操作処理部203、実行制御部204、オブジェクトコード管理部205、シンボル情報管理手段であるシンボル情報管理部206、キャッシュアクセス情報採取手段であるキャッシュアクセス情報採取部207、シンボル配置変更情報生成手段であるシンボル配置変更情報生成部208を備える。

【0051】また、202はシミュレータ104であり、CPUをシミュレーションするCPUシミュレーション部209、内蔵メモリや外付けメモリをシミュレーションするメモリシミュレーション部210、キャッシュメモリをシミュレーションするキャッシュメモリシミュレーション部211、シミュレーション実行の時刻を管理計測する時刻管理部212を備える。

【0052】デバッグ操作処理部203は、オブジェクトコード213のロードや実行開始、ブレークアドレス設定などユーザからのデバッグコマンドの入力を受付

12

け、処理する手段である。例えば、オブジェクトコード213のロードコマンド処理時には、オブジェクトコード管理部205から、オブジェクトコード213を読み込み、シミュレータ202内のメモリシミュレーション部210にオブジェクトコード213の内容を書き込む処理を行う。

【0053】また、実行開始コマンド処理時には、シミュレータ203内のCPUシミュレーション部209のプログラムカウンタ(PC)レジスタに開始アドレスを設定し、実行制御部204により、オブジェクトコードのシミュレーションの実行を開始する。実行制御部204は、シミュレータ202を用いて対象オブジェクトコードの実行を実現する。シミュレーションの実行は、命令を単位として制御可能である。なお、これはプロセッサの動作サイクル毎の制御でもよい。

【0054】この実行制御部204では、ユーザに指定されたブレークポイントにオブジェクトコードの実行が到達するまで、あるいは既定の停止アドレス(例えばexit関数のアドレス)に実行が到達するまで、オブジェクトコード213の実行が繰り返し行われる。図5に、実行制御部204の動作フローチャートを示す。図5において、ステップ801では、シミュレータ202からCPU701(1301)内のプログラムカウンタレジスタ(PC)の値を読み出し、実行アドレスが既定の停止アドレス(例えばexit関数)に到達したかを判定する。到達していれば処理を終了し、到達していなければステップ802に進む。

【0055】ステップ802では、デバッグ操作処理部203から利用者が指定したブレークアドレスの情報を読み出し、実行アドレスがブレークアドレスに到達したかを判定する。到達していれば処理を終了し、到達していなければステップ803に進む。ステップ803では、シミュレーション部202を用いて1命令分のシミュレーションを行う。

【0056】図2において、209はCPUシミュレーション部であり、CPU701(1301)の動作をシミュレーションする。つまり、実行する命令をプログラムカウンタ(PC)レジスタが指し示すアドレスから読み取り(以下フェッチ動作とよぶ)、読み込んだ命令に従ってレジスタあるいはメモリ上のデータを用いて演算を行う。

【0057】フェッチ動作をシミュレーションする場合、メモリシミュレーション部210にアドレス(フェッチアドレス)を入力して、そのアドレス上のデータを得る。また、演算のシミュレーションにおいて、入力としてメモリ上のデータを読み込む(データリード)際には、メモリシミュレーション部210にアドレスを入力し、そのアドレス上のデータを得る。

【0058】同様に演算時に結果をメモリ上に出力する(データライト)際には、メモリシミュレーション部2

10へアドレスと出力するデータ内容を入力し、そのアドレス上のデータへ書き込む。図6に、CPUシミュレーション部209の1命令分のシミュレーションを行う場合の動作フローチャートを示す。図6において、ステップ901では、CPUシミュレーション部209のプログラムカウンタ(PC)レジスタが指し示すアドレスと、1命令の語長である4バイトをメモリシミュレーション部210に指定して、そのアドレス上のデータを読み込む。また、同時に命令の読み込みに必要なサイクル数を求める。

【0059】ステップ902では、入力した機械語命令である4バイト長のビットパターンを解析し、命令の種類および命令実行に必要な入力先および出力先を求める。ビットパターンには、入力元あるいは出力先のレジスタの指定番号や、即値あるいはメモリ上のアドレスが埋込まれている。ステップ903では、命令実行に必要なデータを入力する。データの入力元は命令によって異なり、レジスタの内容やメモリ上の内容あるいは即値である。入力元がレジスタの内容である場合は、CPUシミュレーション部209の中のレジスタ内容管理領域を参照することによって、入力値を得る。また、入力元がメモリ上の内容の場合は、メモリシミュレーション部210に入力元アドレスおよびデータ長を指定することにより、入力値を得る。また、同時にデータの読み込みに必要なサイクル数を求める。また、即値の場合は、ステップ902で解析した結果を参照する。

【0060】ステップ904では、ステップ902で解析した結果を用いて命令の種類を判別し、命令の種類に応じてステップ905a、905b、905c、905d、905e、905f、905g、905h、905iに進む。命令がadd命令の場合は、ステップ905aに進む。ステップ905aでは、add命令の演算シミュレーションを行う。つまりステップ903で入力したデータを用いて加算を実行し、演算結果を得る。

【0061】同様に、命令がinc命令の場合は、ステップ905bに進む。ステップ905bでは、inc命令の演算シミュレーションを行う。つまりステップ903で入力したデータと値1の加算を実行し、演算結果を得る。同様に、命令がsub命令の場合は、ステップ905cに進む。ステップ905cでは、sub命令の演算シミュレーションを行う。つまりステップ903で入力したデータを用いて減算を実行し、演算結果を得る。

【0062】同様に、命令がcmp命令の場合は、ステップ905dに進む。ステップ905dでは、cmp命令の演算シミュレーションを行う。つまりステップ903で入力したデータによる減算を実行し、演算結果フラグを得る。同様に、命令がmul命令の場合は、ステップ905eに進む。ステップ905eでは、mul命令の演算シミュレーションを行う。つまりステップ903で入力したデータを用いて乗算を実行し、演算結果を得

る。

【0063】同様に、命令がclr命令の場合は、ステップ905fに進む。ステップ905fでは、clr命令の演算シミュレーションを行う。つまり演算結果として0を得る。同様に、命令がmov命令の場合は、ステップ905gに進む。ステップ905gでは、mov命令の演算シミュレーションを行う。つまりステップ903で入力したデータを演算結果とする。

【0064】同様に、命令がjmp命令の場合は、ステップ905hに進む。ステップ905hでは、jmp命令の演算シミュレーションを行う。つまりステップ902での解析結果からジャンプ先のアドレスを入力して、CPUシミュレーション部209のプログラムカウンタレジスタに格納する。同様に、命令がbge命令の場合は、ステップ905iに進む。ステップ905iでは、bge命令の演算シミュレーションを行う。つまりCPUシミュレーション部209からプロセッサ状態語レジスタPSWの値を読み出し、フラグを参照して、条件が成立している場合(演算結果フラグが正あるいは0)に、ステップ902での解析結果からジャンプ先のアドレスを入力して、CPUシミュレーション部209のプログラムカウンタレジスタに格納する。

【0065】ステップ905a、905b、905c、905d、905e、905f、905gでは、次に処理終了後ステップ906に進む。ステップ905h、905iでは、次に処理終了後ステップ907に進む。ステップ905a、905b、905c、905d、905e、905f、905g、905hおよび905iでは、それぞれ命令の実行にかかるサイクル数を求める。これは、命令の種類およびその実行時のCPUの状態により決定され、求まる。

【0066】ステップ906では、CPUシミュレーション部209のプログラムカウンタレジスタの値を入力し、値4を加算して格納する。これによりプログラム実行が逐次的に行われる。ステップ907では、ステップ905a、905b、905c、905d、905e、905f、905gで得られた演算結果を、ステップ902での解析で判別した出力先に格納する。演算結果の出力先は、命令によって異なり、レジスタやメモリ上の領域である。出力先がレジスタである場合は、CPUシミュレーション部209の中のレジスタ内容管理領域へ格納する。また、出力先がメモリ上の領域の場合は、メモリシミュレーション部210に出力先アドレスと出力データおよび出力データ長を指定することにより、出力を行う。

【0067】ステップ908では、命令の読み込みにかかるサイクル数(時間)、命令の実行に必要なとなるデータの入力にかかるサイクル数、命令の実行にかかるサイクル数、演算結果の格納にかかるサイクル数など1命令実行に必要なサイクル数を全て算出し、時刻管理部212

へ出力する。図2における時刻管理部212は、1命令ごとのサイクル数を合計していくことにより、実行プログラムのサイクル数を求める。

【0068】図7にメモリシミュレーション部210の動作フローチャートを示す。図7において、ステップ1001では、メモリシミュレーション部210へのアクセスの種別の判定を行う。前述したようにCPUシミュレーション部209からは、命令の読み込み動作（フェッチ動作）の要求、データの読み込み動作（データリード動作）の要求およびデータの書き込み動作（データライト動作）の要求がくる。また、デバグ201からは、デバグコマンドでのメモリ参照操作の場合に、データの読み込み（ダイレクトリード動作）の要求が、またオブジェクトコード213のロード処理やデバグ201上で*

*のメモリ変更操作処理の場合に、データの書き込み（ダイレクトライト動作）の要求がくる。このダイレクトリード動作あるいは、ダイレクトライト動作の場合は、プロセッサのシミュレーションは行わない。つまりプロセッサの状態を変更せず、またサイクル数を算出しない。

【0069】フェッチ動作の場合は、ステップ1002で、アクセス対象として指定されたアドレスがキャッシュابل空間かどうかを判定する。この実施の形態で対象とするプロセッサは、アクセスの対象となるアドレスによってキャッシュが動作するかどうか違い、表4に示すような、区分となっている。

【0070】

【表4】

アドレス範囲	キャッシュ属性
0x00000000-0x3FFFFFFF	非キャッシュابل空間
0x40000000-0x7FFFFFFF	キャッシュابل空間
0x80000000-0xFFFFFFFF	非キャッシュابل空間

【0071】表4は、メモリシミュレーション部210が保持するキャッシュ属性判定表である。表4は、アドレス0x00000000番地から0x3FFFFFFF番地までのアドレス空間およびアドレス0x80000000番地から0xFFFFFFFF番地までのアドレス空間が非キャッシュابل空間であり、アドレス0x40000000番地から0x7FFFFFFF番地までのアドレス空間がキャッシュابل空間であることを示している。キャッシュابل空間の場合、その空間内のアドレスへのアクセスは全てキャッシュメモリを利用して高速化され、非キャッシュابل空間の場合は、キャッシュメモリが利用されない。

【0072】ステップ1002において、指定アドレスがキャッシュابل空間の場合は、ステップ1003に進む。そうでない場合はステップ1004に進む。ステップ1003では、キャッシュメモリシミュレーション部211に対してアドレスと読み込むデータ長を指定することにより、キャッシュメモリのフェッチ動作のシミュレーションを行う。

【0073】ステップ1004では、メモリシミュレーション部210内のメモリデータ格納領域を参照し、指定アドレスのメモリデータを得る。メモリデータ格納領域は図8に示すように、実際のメモリデータとそのアドレスを管理保持する領域である。図8において、1201は、アドレス0x00000000番地から0x00003FFF番地までのアドレス空間のデータ内容を保持管理する領域である。同様に、1202はアドレス0x40000000番地から0x40003FFF番地までのアドレス空間のデータ内容を、1203は0x80000000番地から0x80003FFF番地まで

のアドレス空間のデータ内容を保持管理する領域である。

【0074】ステップ1005では、アクセスにかかるサイクル数を算出する。キャッシュابل空間の場合は、キャッシュメモリシミュレーション部211で算出されたサイクル数を参照する。また非キャッシュابل空間の場合は、シミュレーション対象のメモリのウェイト数およびメモリとプロセッサ間のバス幅などの情報からサイクル数を算出する。

【0075】同様に、データリード動作の場合は、ステップ1006で、アクセス対象として指定されたアドレスがキャッシュابل空間かどうかを判定する。指定アドレスがキャッシュابل空間の場合は、ステップ1007に進む。そうでない場合は、ステップ1008に進む。ステップ1007では、キャッシュメモリシミュレーション部211に対してアドレスと読み込むデータ長を指定することにより、キャッシュメモリのデータリード動作のシミュレーションを行う。

【0076】ステップ1008では、ステップ1004と同様に、メモリシミュレーション部210内のメモリデータ格納領域を参照し、指定アドレスのメモリデータを得る。ステップ1009では、ステップ1005と同様に、アクセスにかかるサイクル数を算出する。

【0077】同様に、データライト動作の場合は、ステップ1010で、アクセス対象として指定されたアドレスがキャッシュابل空間かどうかを判定する。指定アドレスがキャッシュابل空間の場合は、ステップ1011に進む。そうでない場合は、ステップ1012に進む。ステップ1011では、キャッシュメモリシミュレーション部211に対してアドレスと書き込むデータとその

データ長を指定することにより、キャッシュメモリのデータライト動作のシミュレーションを行う。ステップ1012では、メモリシミュレーション部210内のメモリデータ格納領域のデータ内容を変更する。

【0078】ステップ1013では、ステップ1005と同様に、アクセスにかかるサイクル数を算出する。また、ダイレクトリード動作の場合は、ステップ1014において、直接メモリデータ格納領域のデータを参照する。同様に、ダイレクトライト動作の場合は、ステップ1015において、直接メモリデータ格納領域のデータ

10 変更する。
【0079】図10に、キャッシュメモリシミュレーション部211のブロック図を示す。図10において、1401は、キャッシュメモリシミュレーション部であり、命令キャッシュタグ情報管理部1402とデータキャッシュタグ情報管理部1403を備える。命令キャッシュタグ情報管理部1402は、命令キャッシュメモリのシミュレーション時つまりフェッチ動作のシミュレーション時に利用する図11に示すタグ情報管理表を保持し、命令キャッシュメモリ1303上に主記憶装置1302上のどのアドレスのデータが登録されているかを管理する。

【0080】同様に、データキャッシュタグ情報管理部1403は、データキャッシュメモリ1304のシミュレーション時つまりデータリード動作とデータライト動作時のシミュレーション時に利用するタグ情報管理表を保持し、データキャッシュメモリ1304上に登録されている主記憶装置1302上のアドレスを管理する。なお、この実施の形態で対象とするプロセッサでは、キャッシュデータアレイ部自体も、アドレスが割振れてい

30 35 40 45 50 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000 1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055 1060 1065 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115 1120 1125 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175 1180 1185 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235 1240 1245 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295 1300 1305 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355 1360 1365 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415 1420 1425 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475 1480 1485 1490 1495 1500 1505 1510 1515 1520 1525 1530 1535 1540 1545 1550 1555 1560 1565 1570 1575 1580 1585 1590 1595 1600 1605 1610 1615 1620 1625 1630 1635 1640 1645 1650 1655 1660 1665 1670 1675 1680 1685 1690 1695 1700 1705 1710 1715 1720 1725 1730 1735 1740 1745 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085 2090 2095 2100 2105 2110 2115 2120 2125 2130 2135 2140 2145 2150 2155 2160 2165 2170 2175 2180 2185 2190 2195 2200 2205 2210 2215 2220 2225 2230 2235 2240 2245 2250 2255 2260 2265 2270 2275 2280 2285 2290 2295 2300 2305 2310 2315 2320 2325 2330 2335 2340 2345 2350 2355 2360 2365 2370 2375 2380 2385 2390 2395 2400 2405 2410 2415 2420 2425 2430 2435 2440 2445 2450 2455 2460 2465 2470 2475 2480 2485 2490 2495 2500 2505 2510 2515 2520 2525 2530 2535 2540 2545 2550 2555 2560 2565 2570 2575 2580 2585 2590 2595 2600 2605 2610 2615 2620 2625 2630 2635 2640 2645 2650 2655 2660 2665 2670 2675 2680 2685 2690 2695 2700 2705 2710 2715 2720 2725 2730 2735 2740 2745 2750 2755 2760 2765 2770 2775 2780 2785 2790 2795 2800 2805 2810 2815 2820 2825 2830 2835 2840 2845 2850 2855 2860 2865 2870 2875 2880 2885 2890 2895 2900 2905 2910 2915 2920 2925 2930 2935 2940 2945 2950 2955 2960 2965 2970 2975 2980 2985 2990 2995 3000 3005 3010 3015 3020 3025 3030 3035 3040 3045 3050 3055 3060 3065 3070 3075 3080 3085 3090 3095 3100 3105 3110 3115 3120 3125 3130 3135 3140 3145 3150 3155 3160 3165 3170 3175 3180 3185 3190 3195 3200 3205 3210 3215 3220 3225 3230 3235 3240 3245 3250 3255 3260 3265 3270 3275 3280 3285 3290 3295 3300 3305 3310 3315 3320 3325 3330 3335 3340 3345 3350 3355 3360 3365 3370 3375 3380 3385 3390 3395 3400 3405 3410 3415 3420 3425 3430 3435 3440 3445 3450 3455 3460 3465 3470 3475 3480 3485 3490 3495 3500 3505 3510 3515 3520 3525 3530 3535 3540 3545 3550 3555 3560 3565 3570 3575 3580 3585 3590 3595 3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650 3655 3660 3665 3670 3675 3680 3685 3690 3695 3700 3705 3710 3715 3720 3725 3730 3735 3740 3745 3750 3755 3760 3765 3770 3775 3780 3785 3790 3795 3800 3805 3810 3815 3820 3825 3830 3835 3840 3845 3850 3855 3860 3865 3870 3875 3880 3885 3890 3895 3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965 3970 3975 3980 3985 3990 3995 4000 4005 4010 4015 4020 4025 4030 4035 4040 4045 4050 4055 4060 4065 4070 4075 4080 4085 4090 4095 4100 4105 4110 4115 4120 4125 4130 4135 4140 4145 4150 4155 4160 4165 4170 4175 4180 4185 4190 4195 4200 4205 4210 4215 4220 4225 4230 4235 4240 4245 4250 4255 4260 4265 4270 4275 4280 4285 4290 4295 4300 4305 4310 4315 4320 4325 4330 4335 4340 4345 4350 4355 4360 4365 4370 4375 4380 4385 4390 4395 4400 4405 4410 4415 4420 4425 4430 4435 4440 4445 4450 4455 4460 4465 4470 4475 4480 4485 4490 4495 4500 4505 4510 4515 4520 4525 4530 4535 4540 4545 4550 4555 4560 4565 4570 4575 4580 4585 4590 4595 4600 4605 4610 4615 4620 4625 4630 4635 4640 4645 4650 4655 4660 4665 4670 4675 4680 4685 4690 4695 4700 4705 4710 4715 4720 4725 4730 4735 4740 4745 4750 4755 4760 4765 4770 4775 4780 4785 4790 4795 4800 4805 4810 4815 4820 4825 4830 4835 4840 4845 4850 4855 4860 4865 4870 4875 4880 4885 4890 4895 4900 4905 4910 4915 4920 4925 4930 4935 4940 4945 4950 4955 4960 4965 4970 4975 4980 4985 4990 4995 5000 5005 5010 5015 5020 5025 5030 5035 5040 5045 5050 5055 5060 5065 5070 5075 5080 5085 5090 5095 5100 5105 5110 5115 5120 5125 5130 5135 5140 5145 5150 5155 5160 5165 5170 5175 5180 5185 5190 5195 5200 5205 5210 5215 5220 5225 5230 5235 5240 5245 5250 5255 5260 5265 5270 5275 5280 5285 5290 5295 5300 5305 5310 5315 5320 5325 5330 5335 5340 5345 5350 5355 5360 5365 5370 5375 5380 5385 5390 5395 5400 5405 5410 5415 5420 5425 5430 5435 5440 5445 5450 5455 5460 5465 5470 5475 5480 5485 5490 5495 5500 5505 5510 5515 5520 5525 5530 5535 5540 5545 5550 5555 5560 5565 5570 5575 5580 5585 5590 5595 5600 5605 5610 5615 5620 5625 5630 5635 5640 5645 5650 5655 5660 5665 5670 5675 5680 5685 5690 5695 5700 5705 5710 5715 5720 5725 5730 5735 5740 5745 5750 5755 5760 5765 5770 5775 5780 5785 5790 5795 5800 5805 5810 5815 5820 5825 5830 5835 5840 5845 5850 5855 5860 5865 5870 5875 5880 5885 5890 5895 5900 5905 5910 5915 5920 5925 5930 5935 5940 5945 5950 5955 5960 5965 5970 5975 5980 5985 5990 5995 6000 6005 6010 6015 6020 6025 6030 6035 6040 6045 6050 6055 6060 6065 6070 6075 6080 6085 6090 6095 6100 6105 6110 6115 6120 6125 6130 6135 6140 6145 6150 6155 6160 6165 6170 6175 6180 6185 6190 6195 6200 6205 6210 6215 6220 6225 6230 6235 6240 6245 6250 6255 6260 6265 6270 6275 6280 6285 6290 6295 6300 6305 6310 6315 6320 6325 6330 6335 6340 6345 6350 6355 6360 6365 6370 6375 6380 6385 6390 6395 6400 6405 6410 6415 6420 6425 6430 6435 6440 6445 6450 6455 6460 6465 6470 6475 6480 6485 6490 6495 6500 6505 6510 6515 6520 6525 6530 6535 6540 6545 6550 6555 6560 6565 6570 6575 6580 6585 6590 6595 6600 6605 6610 6615 6620 6625 6630 6635 6640 6645 6650 6655 6660 6665 6670 6675 6680 6685 6690 6695 6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750 6755 6760 6765 6770 6775 6780 6785 6790 6795 6800 6805 6810 6815 6820 6825 6830 6835 6840 6845 6850 6855 6860 6865 6870 6875 6880 6885 6890 6895 6900 6905 6910 6915 6920 6925 6930 6935 6940 6945 6950 6955 6960 6965 6970 6975 6980 6985 6990 6995 7000 7005 7010 7015 7020 7025 7030 7035 7040 7045 7050 7055 7060 7065 7070 7075 7080 7085 7090 7095 7100 7105 7110 7115 7120 7125 7130 7135 7140 7145 7150 7155 7160 7165 7170 7175 7180 7185 7190 7195 7200 7205 7210 7215 7220 7225 7230 7235 7240 7245 7250 7255 7260 7265 7270 7275 7280 7285 7290 7295 7300 7305 7310 7315 7320 7325 7330 7335 7340 7345 7350 7355 7360 7365 7370 7375 7380 7385 7390 7395 7400 7405 7410 7415 7420 7425 7430 7435 7440 7445 7450 7455 7460 7465 7470 7475 7480 7485 7490 7495 7500 7505 7510 7515 7520 7525 7530 7535 7540 7545 7550 7555 7560 7565 7570 7575 7580 7585 7590 7595 7600 7605 7610 7615 7620 7625 7630 7635 7640 7645 7650 7655 7660 7665 7670 7675 7680 7685 7690 7695 7700 7705 7710 7715 7720 7725 7730 7735 7740 7745 7750 7755 7760 7765 7770 7775 7780 7785 7790 7795 7800 7805 7810 7815 7820 7825 7830 7835 7840 7845 7850 7855 7860 7865 7870 7875 7880 7885 7890 7895 7900 7905 7910 7915 7920 7925 7930 7935 7940 7945 7950 7955 7960 7965 7970 7975 7980 7985 7990 7995 8000 8005 8010 8015 8020 8025 8030 8035 8040 8045 8050 8055 8060 8065 8070 8075 8080 8085 8090 8095 8100 8105 8110 8115 8120 8125 8130 8135 8140 8145 8150 8155 8160 8165 8170 8175 8180 8185 8190 8195 8200 8205 8210 8215 8220 8225 8230 8235 8240 8245 8250 8255 8260 8265 8270 8275 8280 8285 8290 8295 8300 8305 8310 8315 8320 8325 8330 8335 8340 8345 8350 8355 8360 8365 8370 8375 8380 8385 8390 8395 8400 8405 8410 8415 8420 8425 8430 8435 8440 8445 8450 8455 8460 8465 8470 8475 8480 8485 8490 8495 8500 8505 8510 8515 8520 8525 8530 8535 8540 8545 8550 8555 8560 8565 8570 8575 8580 8585 8590 8595 8600 8605 8610 8615 8620 8625 8630 8635 8640 8645 8650 8655 8660 8665 8670 8675 8680 8685 8690 8695 8700 8705 8710 8715 8720 8725 8730 8735 8740 8745 8750 8755 8760 8765 8770 8775 8780 8785 8790 8795 8800 8805 8810 8815 8820 8825 8830 8835 8840 8845 8850 8855 8860 8865 8870 8875 8880 8885 8890 8895 8900 8905 8910 8915 8920 8925 8930 8935 8940 8945 8950 8955 8960 8965 8970 8975 8980 8985 8990 8995 9000 9005 9010 9015 9020 9025 9030 9035 9040 9045 9050 9055 9060 9065 9070 9075 9080 9085 9090 9095 9100 9105 9110 9115 9120 9125 9130 9135 9140 9145 9150 9155 9160 9165 9170 9175 9180 9185 9190 9195 9200 9205 9210 9215 9220 9225 9230 9235 9240 9245 9250 9255 9260 9265 9270 9275 9280 9285 9290 9295 9300 9305 9310 9315 9320 9325 9330 9335 9340 9345 9350 9355 9360 9365 9370 9375 9380 9385 9390 9395 9400 9405 9410 9415 9420 9425 9430 9435 9440 9445 9450 9455 9460 9465 9470 9475 9480 9485 9490 9495 9500 9505 9510 9515 9520 9525 9530 9535 9540 9545 9550 9555 9560 9565 9570 9575 9580 9585 9590 9595 9600 9605 9610 9615 9620 9625 9630 9635 9640 9645 9650 9655 9660 9665 9670 9675 9680 9685 9690 9695 9700 9705 9710 9715 9720 9725 9730 9735 9740 9745 9750 9755 9760 9765 9770 9775 9780 9785 9790 9795 9800 9805 9810 9815 9820 9825 9830 9835 9840 9845 9850 9855 9860 9865 9870 9875 9880 9885 9890 9895 9900 9905 9910 9915 9920 9925 9930 9935 9940 9945 9950 9955 9960 9965 9970 9975 9980 9985 9990 9995 10000 10005 10010 10015 10020 10025 10030 10035 10040 10045 10050 10055 10060 10065 10070 10075 10080 10085 10090 10095 10100 10105 10110 10115 10120 10125 10130 10135 10140 10145 10150 10155 10160 10165 10170 10175 10180 10185 10190 10195 10200 10205 10210 10215 10220 10225 10230 10235 10240 10245 10250 10255 10260 10265 10270 10275 10280 10285 10290 10295 10300 10305 10310 10315 10320 10325 10330 10335 10340 10345 10350 10355 10360 10365 10370 10375 10380 10385 10390 10395 10400 10405 10410 10415 10420 10425 10430 10435 10440 10445 10450 10455 10460 10465 10470 10475 10480 10485 10490 10495 10500 10505 10510 10515 10520 10525 10530 10535 10540 10545 10550 10555 10560 10565 10570 10575 10580 10585 10590 10595 10600 10605 10610 10615 10620 10625 10630 10635 10640 10645 10650 10655 10660 10665 10670 10675 10680 10685 10690 10695 10700 10705 10710 10715 10720 10725 10730 10735 10740 10745 10750 10755 10760 10765 10770 10775 10780 10785 10790 10795 10800 10805 10810 10815 10820 10825 10830 10835 10840 10845 10850 10855 10860 10865 10870 10875 10880 10885 10890 10895 10900 10905 10910 10915 10920 10925 10930 10935 10940 10945 10950 10955 10960 10965 10970 10975 10980 10985 10990 10995 11000 11005 11010 11015 11020 11025 11030 11035 11040 11045 11050 11055 11060 11065 11070 11075 11080 11085 11090 11095 11100 11105 11110 11115 11120 11125 11130 11135 11140 11145 11150 11155 11160 11165 11170 11175 11180 11185 11190 11195 11200 11205 11210 11215 11220 11225 11230 11235 11240 11245 11250 11255 11260 11265 11270 11275 11280 11285 11290 11295 11300 11305 11310 11315 11320 11325 11330 11335 11340 11345 11350 11355 11360 11365 11370 11375 11380 11385 11390 11395 11400 11405 11410 11415 11420 11425 11430 11435 11440 11445 11450 11455 11460 11465 11470 11475 11480 11485 11490 11495 11500 11505 11510 11515 11520 11525 11530 11535 11540 11545 11550 11555 11560 11565 11570 11575 11580 11585 11590 11595 11600 11605 11610 11615 11620 11625 11630 11635 116

プ1613に進む。

【0087】ステップ1612では、取得したタグ情報と指定アドレスを比較し、キャッシュヒットかどうかを判定する。キャッシュミスと判定された場合は、競合性ミスであり、この場合競合性ミスであることを記憶する。なお、競合性ミスであることはステップ1615において、キャッシュアクセス情報採取部207に登録される。

【0088】キャッシュミスの場合、ステップ1613に進む。ステップ1613では、エントリにそのアドレスを登録する。ステップ1614では、データリード動作におけるデータキャッシュミスのサイクル数を求める。ステップ1615では、データキャッシュへのデータリード動作時のアクセスがキャッシュミスであったこと、ならびにアクセスアドレス、データ長、およびエントリをキャッシュアクセス情報採取部207に登録する。また、CPUシミュレーション部209からプログラムカウンタレジスタ(PC)の値と、スタックポインタレジスタ(SP)の値を取得し、共に登録する。また、競合性ミスであったかどうか登録する。

【0089】ステップ1616では、データリード動作におけるデータキャッシュヒット時のサイクル数を求める。ステップ1617では、データキャッシュへのデータリード動作時のアクセスがキャッシュヒットであったことをキャッシュアクセス情報採取部207に登録する。またアクセスアドレス、データ長、およびエントリをキャッシュアクセス情報採取部207に登録する。また、CPUシミュレーション部209からプログラムカウンタレジスタ(PC)の値と、スタックポインタレジスタ(SP)の値を取得し、共に登録する。

【0090】同様に、動作がデータライト動作の場合は、ステップ1618に進み、データキャッシュメモリ上のエントリを指定されたアドレスから算出する。ステップ1619では、エントリのタグ情報および有効フラグをデータキャッシュタグ情報管理部1403から検索し、有効フラグを判定する。そのエントリが有効な場合は、ステップ1620に進み、無効の場合はステップ1621に進む。

【0091】ステップ1620では、取得したタグ情報と指定アドレスを比較し、キャッシュヒットかどうかを判定する。キャッシュミスと判定された場合は、競合性ミスであり、この場合競合性ミスであることを記憶する。なお、競合性ミスであることはステップ1623において、キャッシュアクセス情報採取部207に登録される。

【0092】キャッシュミスの場合、ステップ1621に進む。ステップ1621では、エントリにそのアドレスを登録する。ステップ1622では、データライト動作におけるデータキャッシュミスのサイクル数を求める。ステップ1623では、データキャッシュへのデー

タライト動作時のアクセスがキャッシュミスであったこと、ならびにアクセスアドレス、データ長、およびエントリをキャッシュアクセス情報採取部207に登録する。また、CPUシミュレーション部209からプログラムカウンタレジスタ(PC)の値と、スタックポインタレジスタ(SP)の値を取得し、共に登録する。また、競合性ミスであったかどうか登録する。

【0093】ステップ1624では、データライト動作におけるデータキャッシュヒット時のサイクル数を求める。ステップ1625では、データキャッシュへのデータライト動作時のアクセスがキャッシュヒットであったことをキャッシュアクセス情報採取部207に登録する。また、アクセスアドレス、データ長、およびエントリをキャッシュアクセス情報採取部207に登録する。また、CPUシミュレーション部209からプログラムカウンタレジスタ(PC)の値と、スタックポインタレジスタ(SP)の値を取得し、共に登録する。

【0094】図13に、キャッシュアクセス情報採取部207のブロック図を示す。図13において、1701で示すキャッシュアクセス情報採取部207は、命令キャッシュアクセス情報管理部1702と、データキャッシュアクセス情報管理部1703を備える。命令キャッシュアクセス情報管理部1702は、フェッチアクセス情報1704を保持する。フェッチアクセス情報1704は、キャッシュメモリでのエントリ番号、フェッチアクセスにおけるアクセスアドレスとそのアクセスがキャッシュヒットであったかどうかの情報、そのデータ長、また対応するシンボルとそのアクセスの回数およびキャッシュミスの場合はその内の競合性ミスの回数を含む情報である。

【0095】また同様に、データキャッシュアクセス情報管理部1703は、データリードアクセス情報1705とデータライトアクセス情報1706を保持する。データリードアクセス情報1705は、データリードアクセスにおけるアクセスアドレスとそのアクセスがキャッシュヒットであったかどうかの情報、そのデータ長、また対応するシンボルとそのアクセスの回数、およびキャッシュミスの場合はその内の競合性ミスの回数を含む情報である。

【0096】同様に、データライトアクセス情報1706は、データライトアクセスにおけるアクセスアドレスとそのアクセスがキャッシュヒットであったかどうかの情報、そのデータ長、また対応するシンボルとそのアクセスの回数、およびキャッシュミスの場合はその内の競合性ミスの回数を含む情報である。図14に、キャッシュアクセス情報採取部207の動作フローチャートを示す。図14において、ステップ1801では、キャッシュメモリシミュレーション部1401から入力された採取情報の種別の判定を行う。採取情報が、フェッチ動作に関するものならばステップ1802に進み、データリ

ード動作に関するものならばステップ1804に進み、データライト動作に関するものならばステップ1806に進む。

【0097】ステップ1802では、フェッチアクセスの対象アドレスをシンボル情報から検索し、シンボルとの関連を付ける。すなわち、アドレスをキーにシンボル情報を検索し、シンボル名を得る。ステップ1803では、フェッチアクセス情報1704に、アクセス対象のアドレス、キャッシュヒットか否か、アクセスデータ長、対応付けられたシンボルを登録する。既に同一の情報10が登録されていればそのアクセスの回数の値を増やす。

【0098】ステップ1804では、データリードアクセスの対象アドレスをシンボル情報から検索し、シンボルとの関連を付ける。すなわち、アドレスをキーにシンボル情報を検索し、シンボル名を得る。また、その際にシンボル情報中で局所変数のように関数とスタックポインタレジスタ(SP)からのオフセットして定義されているシンボル情報は、前記の採取情報に含まれているアクセス時のプログラムカウンタ(PC)から関数が判別20でき、さらにアクセス時のスタックポインタレジスタ(SP)の値からシンボルとの対応付けができる。

【0099】ステップ1805では、データリードアクセス情報1705に、アクセス対象のアドレス、キャッシュヒットか否か、アクセスデータ長、対応付けられたシンボルを登録する。既に同一の情報が登録されていればそのアクセスの回数の値を増やす。同様に、ステップ1806では、データライトアクセスの対象アドレスをシンボル情報から検索し、シンボルとの関連を付ける。すなわち、アドレスをキーにシンボル情報を検索し、シンボル名を得る。また、その際にシンボル情報中で局所変数のように関数とスタックポインタレジスタ(SP)からのオフセットして定義されているシンボル情報は、前記の採取情報に含まれているアクセス時のプログラム変数x: 0x0400 → 0x1140 * 21

のように表示する。

【0104】データキャッシュに関して同様に、ステップ1905から1908まで、それぞれステップ1901から1904に対応する処理を行う。ステップ1905において、ステップ1901と同様に、データキャッシュに関する情報であるデータリードアクセス情報1705およびデータライトアクセス情報1706について、各エントリの情報を走査し、図16(b)に示すデータキャッシュ競合情報表2102に登録する。ステップ1906において、データキャッシュ競合情報管理表2102から競合ミスをおこしたシンボルを検索する。シンボルがあればステップ1907に進み、シンボルがなければ処理を終了する。例えば、競合するシンボルとしては、『変数x』と『変数i@main』が得られ、

*カウンタ(PC)から関数が判別でき、さらにアクセス時のスタックポインタレジスタ(SP)の値からシンボルとの対応付けができる。

【0100】ステップ1807では、データライトアクセス情報1706に、アクセス対象のアドレス、キャッシュヒットか否か、アクセスデータ長、および、対応付けられたシンボルを登録する。既に同一の情報が登録されていればそのアクセスの回数の値を増やす。なお、シンボル名が対応付かない場合は、シンボル名なしとして登録する。

【0101】図15に、プログラムのシンボルの配置を変更するためのシンボル配置変更情報生成部208の動作フローチャートを示す。ステップ1901において、命令キャッシュに関する情報であるフェッチアクセス情報1704について、各エントリの情報を走査し、競合性ミスが発生しなかったシンボル、競合性ミスが発生したシンボルをそれぞれ図16(a)に示す命令キャッシュ競合情報表2101に登録する。

【0102】ステップ1902において、命令キャッシュ競合情報管理表2101から競合ミスをおこしたシンボルを検索する。シンボルがあればステップ1903に進み、シンボルがなければステップ1905に進む。ステップ1903において、命令キャッシュ競合情報管理表2101からキャッシュがヒットしていない空のエントリを検索する。空のエントリがあればステップ1904に進み、空のエントリがなければステップ1905に進む。

【0103】ステップ1904において、競合ミスをおこしたシンボルを空のエントリ割付け、対応するアドレスを生成する。空のエントリ番号からアドレスの変更は、例えばある既定値からのエントリ番号に対応するオフセットを足し込むなどして求めることが可能である。生成する情報であるシンボル配置変更情報は、たとえば

【0105】ステップ1907において、データキャッシュ競合情報管理表2102からキャッシュがヒットしていない空のエントリを検索する。空のエントリがあればステップ1908に進み、空のエントリがなければ処理を終了する。例えば、空のエントリとして0x1140が得られる。ステップ1908において、競合ミスをおこしたシンボルを空のエントリ割付け、対応するアドレスを生成する。

【0106】上記の表示(1)に、シンボル配置変更情報生成部208によるシンボル配置変更情報の一例を示す。表示(1)は、シンボルxをアドレス0x0400から0x1140へ変更することを示している。以上のように、この第1の実施の形態によれば、キャッシュメモリへのアクセスの状況を高級言語のシンボルとして検出し、キャッシュメモリで競合性ミスが発生するシンボ

ルの配置を変更する情報が生成できる。

【0107】この発明の第2の実施の形態を図17から図19により説明する。すなわち、このソフトウェア開発支援システムは、第1の実施の形態の図2に示すシンボル配置変更情報生成部208に代えて、表示手段であるキャッシュアクセス情報表示部216を設けたものである。図18に、キャッシュアクセス情報表示部216の動作フローチャートを示す。ステップ1901において、命令キャッシュに関する情報であるフェッチアクセス情報をエントリ別にわけて表示する。ステップ1902において、データキャッシュに関する情報であるデータリードアクセス情報とデータライトアクセス情報とをエントリ別にわけて表示する。

【0108】図19にキャッシュアクセス情報表示部216によるキャッシュアクセス情報の一例を示す。図19において、エントリおよびシンボル別にデータリード動作、データライト動作のヒットとミスの頻度の表示を示している。このように、キャッシュメモリへのアクセスの状況を高級言語レベルで利用者に表示することができる。

【0109】第2の実施の形態によれば、キャッシュアクセス情報採取部207で採取された情報と、シンボル情報管理部206で管理されているシンボル情報をキャッシュアクセス情報表示部216により関連付けて表示することができるので、キャッシュメモリへのアクセスの状況が、対象プログラムが記述される高級言語レベルで表示可能となり、評価性能に基いたプログラムの変更が容易になる。したがって、キャッシュメモリ搭載のプログラムを開発し評価する際に、キャッシュメモリへのアクセス状況を高級言語レベルで知ることができ、プログラム開発の効率化が図れる。

【0110】その他は、第1の実施の形態と同様である。なお、他の実施の形態として、第1の実施の形態に第2の実施の形態のキャッシュアクセス情報表示部216を付加したソフトウェア開発支援システムも可能である。また、この発明において、キャッシュアクセス情報を採取する区間を指定する手段を設け、指定された区間についてのみキャッシュアクセスに関する情報を採取することにより、高速性が求められるプログラムの特定の区間のみについて、性能向上を図ることも可能である。

【0111】

【発明の効果】請求項1記載のソフトウェア開発支援システムによれば、キャッシュアクセス情報採取手段で採取された情報と、シンボル情報管理手段で管理されているシンボル情報から、プログラムのメモリアクセス時の速度が高速になるようにシンボルの配置を変更するためのシンボルの配置変更情報を生成することができる。これにより、キャッシュメモリへのアクセスの状況が対象プログラムが記述される高級言語レベルで検知可能となり、アクセス速度を低下させるシンボルの配置を変更

し、プログラムの実行性能を向上させることが可能となる。

【0112】請求項2記載のソフトウェア開発システムによれば、請求項1と同様な効果のほか、キャッシュアクセス情報採取手段で採取された情報からキャッシュ競合性ミスを検出し、シンボル情報と関連づけて、シンボルの配置変更情報をシンボル配置変更情報生成手段で生成することができる。これにより、キャッシュメモリ搭載のプログラムを開発する際に、キャッシュメモリへのアクセスの状況特に競合性ミスをシンボルレベルないし高級言語レベルで検出し、キャッシュメモリで競合性ミスが発生するシンボルの配置を変更する情報を生成するなど、その競合を解消するためにシンボルの配置を変更することが容易にでき、プログラム開発の効率化が図れる。

【0113】請求項3記載のソフトウェア開発システムによれば、請求項1または請求項2と同様な効果のほか、競合性のキャッシュミスが発生したシンボルをキャッシュされていない空のエントリに対応するアドレスに変更するための情報を生成するなど、キャッシュメモリの競合性ミスを低減させるようなシンボルの配置変更情報が生成でき、より一層のプログラムの性能向上を図れる。

【0114】請求項4記載のソフトウェア開発システムによれば、キャッシュアクセス情報採取手段で採取された情報と、シンボル情報管理手段で管理されているシンボル情報を表示手段により関連付けて表示することができるので、キャッシュメモリへのアクセスの状況が、対象プログラムが記述される高級言語レベルで表示可能となり、評価性能に基いたプログラムの変更が容易になる。したがって、キャッシュメモリ搭載のプログラムを開発し評価する際に、キャッシュメモリへのアクセス状況を高級言語レベルで知ることができ、プログラム開発の効率化が図れる。

【0115】請求項5記載のソフトウェア開発システムによれば、請求項4と同様な効果がある。請求項6記載のソフトウェア開発システムによれば、請求項4または請求項5と同様な効果がある。

【図面の簡単な説明】

【図1】この発明の第1の実施の形態におけるソフトウェア開発支援システムに適用されるシステムのブロック図である。

【図2】この発明の第1の実施の形態におけるソフトウェア開発支援システムのブロック図である。

【図3】キャッシュメモリ搭載プロセッサのシステムブロック図である。

【図4】プロセッサの動作モデルを説明する説明図である。

【図5】図2の実行制御部204の動作フローチャートである。

【図6】図2のCPUシミュレーション部209の動作フローチャートである。

【図7】図2のメモリシミュレーション部210の動作フローチャートである。

【図8】メモリシミュレーション部のメモリデータ格納領域の説明図である。

【図9】この実施の形態におけるキャッシュメモリ搭載プロセッサのシステムブロック図である。

【図10】キャッシュシミュレーション部のブロック図である。

【図11】キャッシュタグ情報管理表を示す説明図である。

【図12】キャッシュシミュレーション部211の動作フローチャートである。

【図13】キャッシュアクセス情報採取部のブロック図である。

【図14】キャッシュアクセス情報採取部の動作フローチャートである。

【図15】シンボル配置変更情報生成部の動作フローチャートである。

【図16】キャッシュ競合情報表を示すブロック図である。

【図17】第2の実施の形態におけるソフトウェア開発支援システムのブロック図である。

【図18】キャッシュアクセス情報表示部の動作フローチャートである。

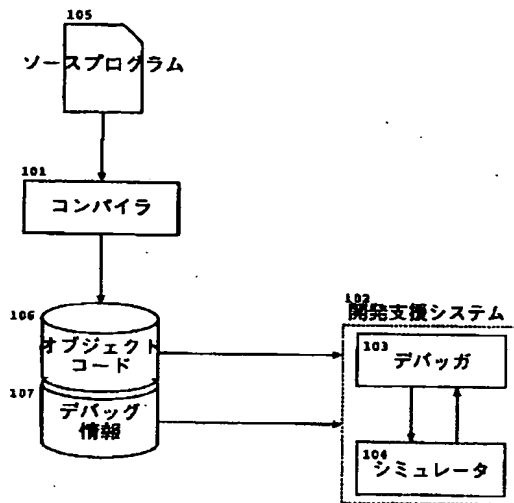
【図19】キャッシュアクセス情報の表示例を示す説明図である。

【符号の説明】

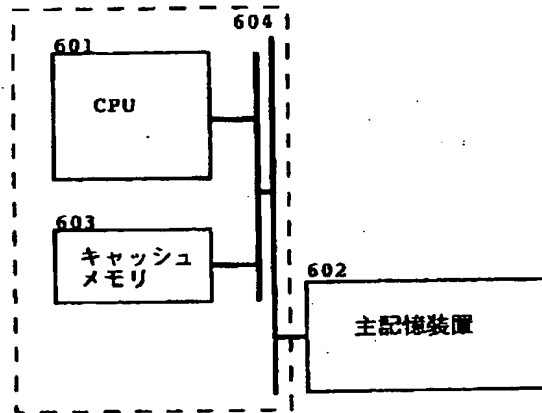
101 コンパイラ
102 開発支援システム
103 デバッガ
104 シミュレータ
105 ソースプログラム
106 オブジェクトコード
107 デバッグ情報
201 デバッガ
202 シミュレータ
203 デバッグ操作処理部
204 実行制御部
205 オブジェクトコード管理部
206 シンボル情報管理部
207 キャッシュアクセス情報採取部
208 シンボル配置変更情報生成部
209 CPUシミュレーション部
210 メモリシミュレーション部

211 キャッシュメモリシミュレーション部
212 時刻管理部
216 キャッシュアクセス情報表示部
601 CPU
602 主記憶装置
603 キャッシュメモリ
604 バス
701 CPU
702 メモリ
703、704、705、706 データレジスタ
707、708、709、710 アドレスレジスタ
711 プロセッサ状態語レジスタ
712 プログラムカウンタレジスタ
713 スタックポインタレジスタ
714 演算器
715 実行コード格納領域
716 データ格納領域
717 スタック領域
1201、1202、1203 アドレス空間メモリデータ格納領域
1301 CPU
1302 主記憶装置
1303 命令キャッシュメモリ
1304 データキャッシュメモリ
1305 バス
1306 命令キャッシュメモリのタグアレイ
1307 命令キャッシュメモリのデータアレイ
1308 データキャッシュメモリのタグアレイ
1309 データキャッシュメモリのデータアレイ
1401 キャッシュメモリシミュレーション部
1402 命令キャッシュタグ情報管理部
1403 データキャッシュタグ情報管理部
1501 有効フラグ
1502 エントリ
1503 タグ
1504 有効なエントリの一例
1505 無効なエントリの一例
1701 キャッシュアクセス情報採取部
1702 命令キャッシュアクセス情報管理部
1703 データキャッシュアクセス情報管理部
1704 フェッチアクセス情報
1705 データリードアクセス情報
1706 データライトアクセス情報
2101 命令キャッシュ競合情報管理表
2102 データキャッシュ競合情報管理表

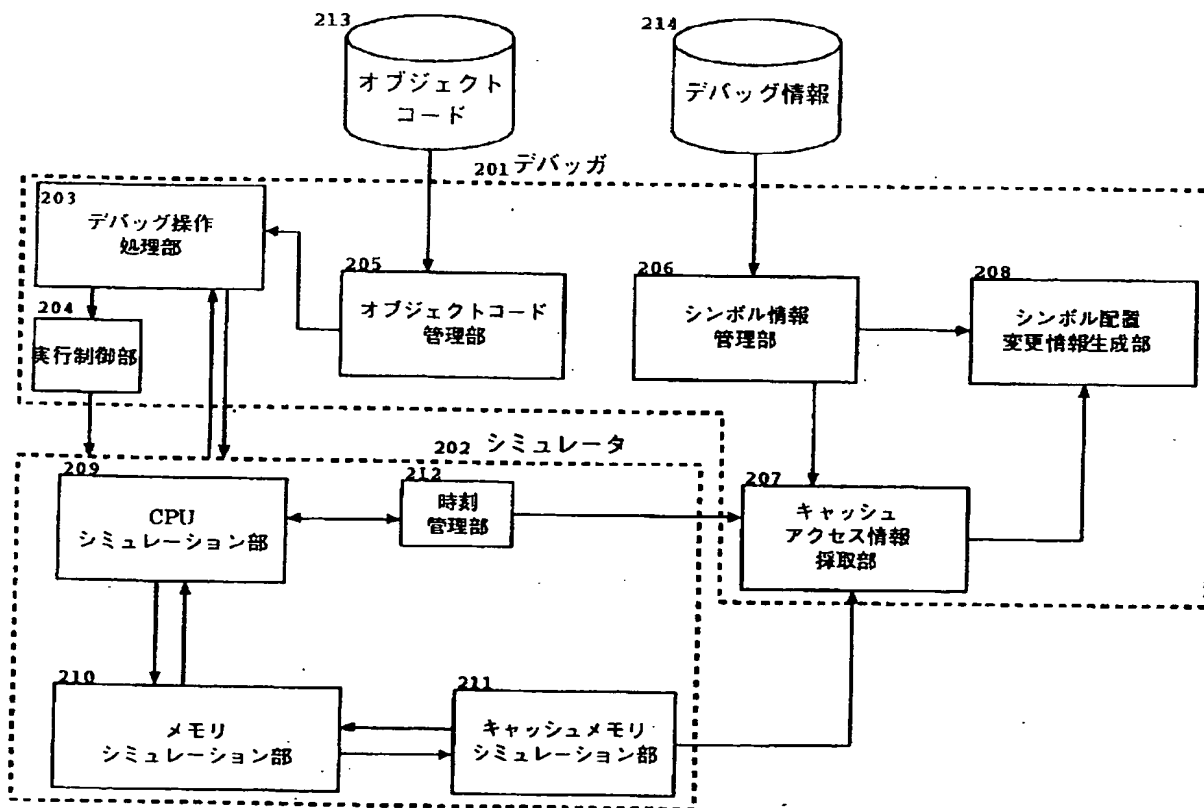
【図1】



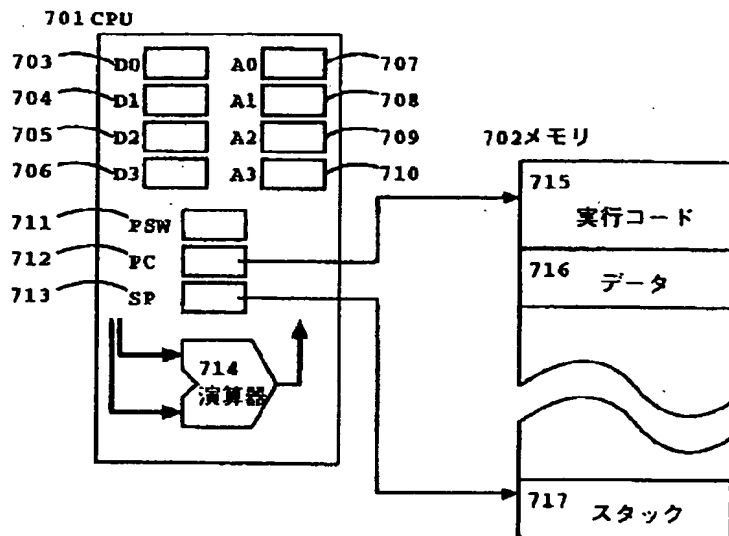
【図3】



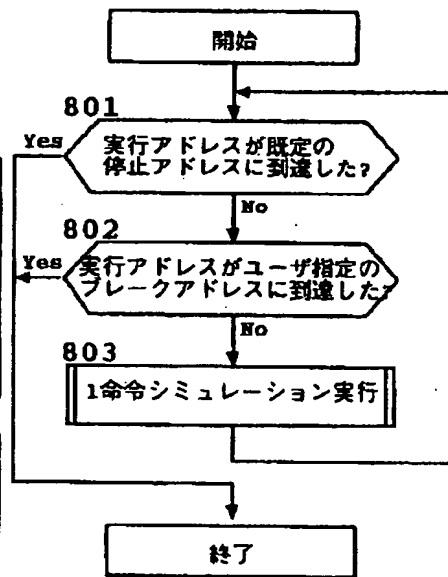
【図2】



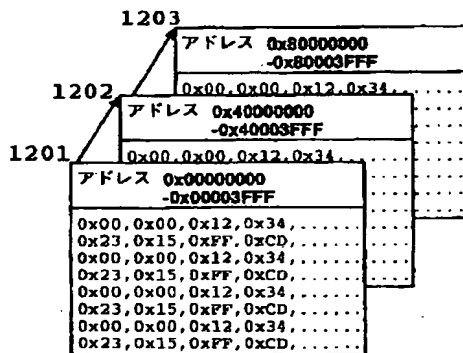
【図4】



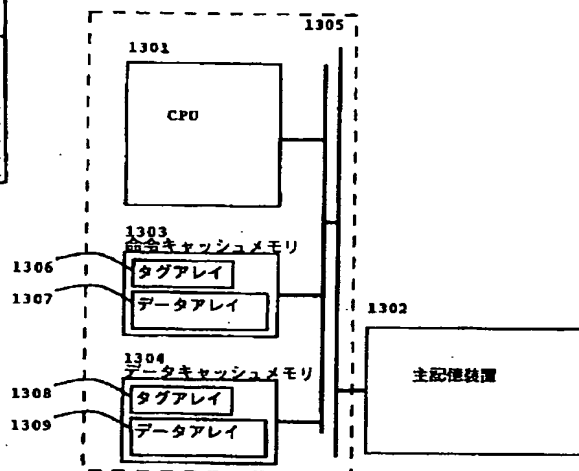
【図5】



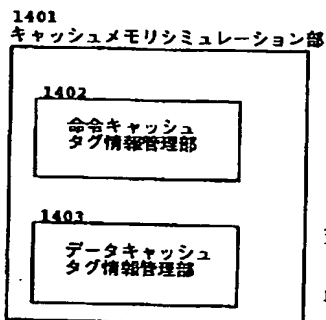
【図8】



【図9】



【図10】



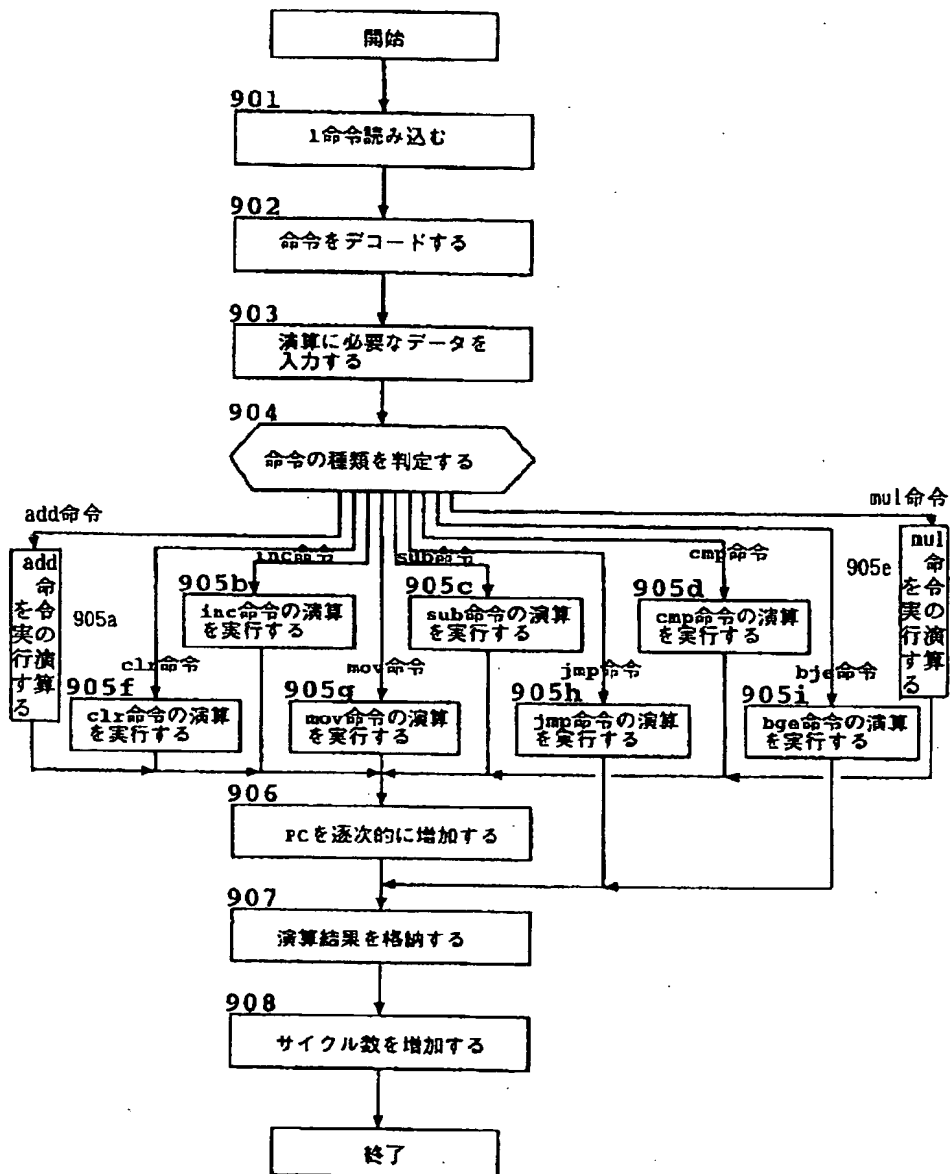
【図11】

エントリ	有効	タグ
0	○	0x40001000
1	○	0x40001010
2	—	
3	○	0x40002020
...
126	○	0x400007E0
127	○	0x400017F0

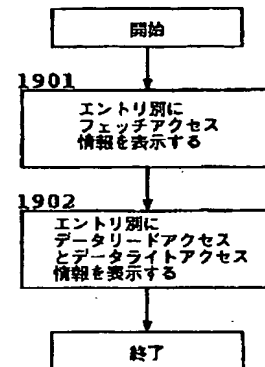
【図19】

エントリ	シンボル	ヒット		ミス	
		リード	ライト	リード	ライト
0x40	変数x	0	0	0	101
	変数1@main	300	100	100	1
0x41	変数y	100	100	0	1

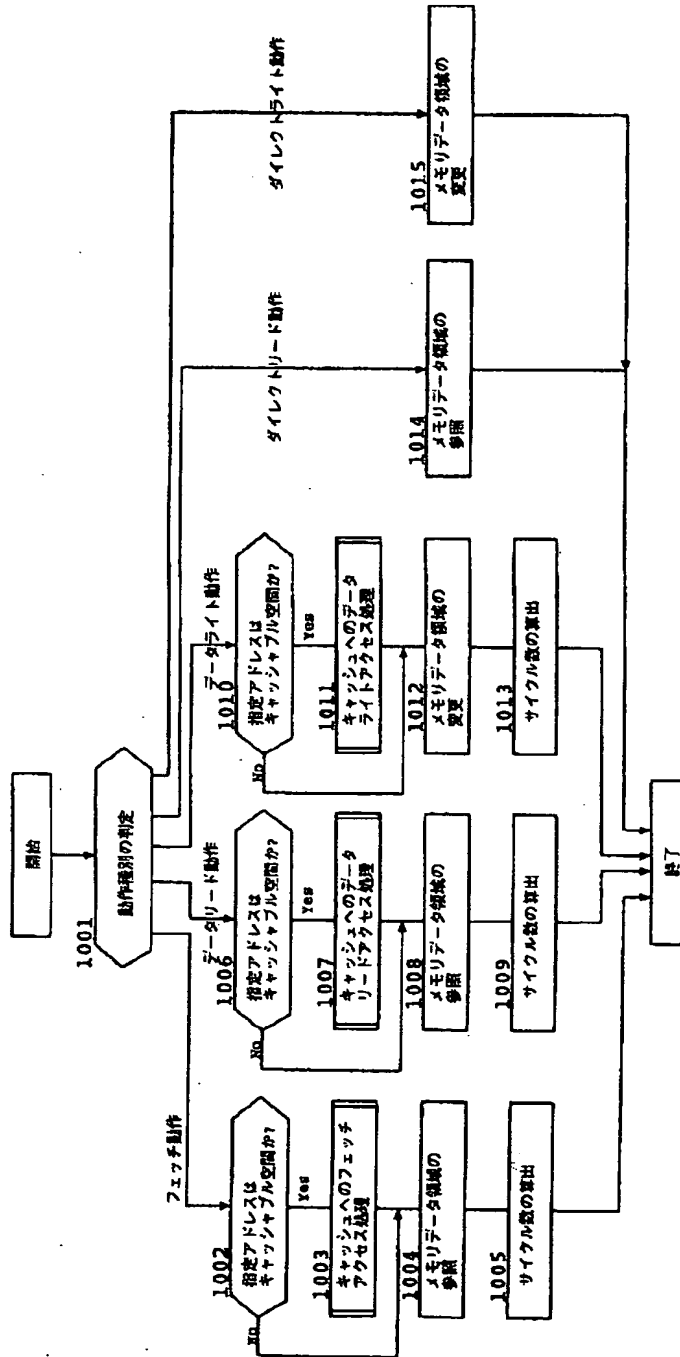
【図6】



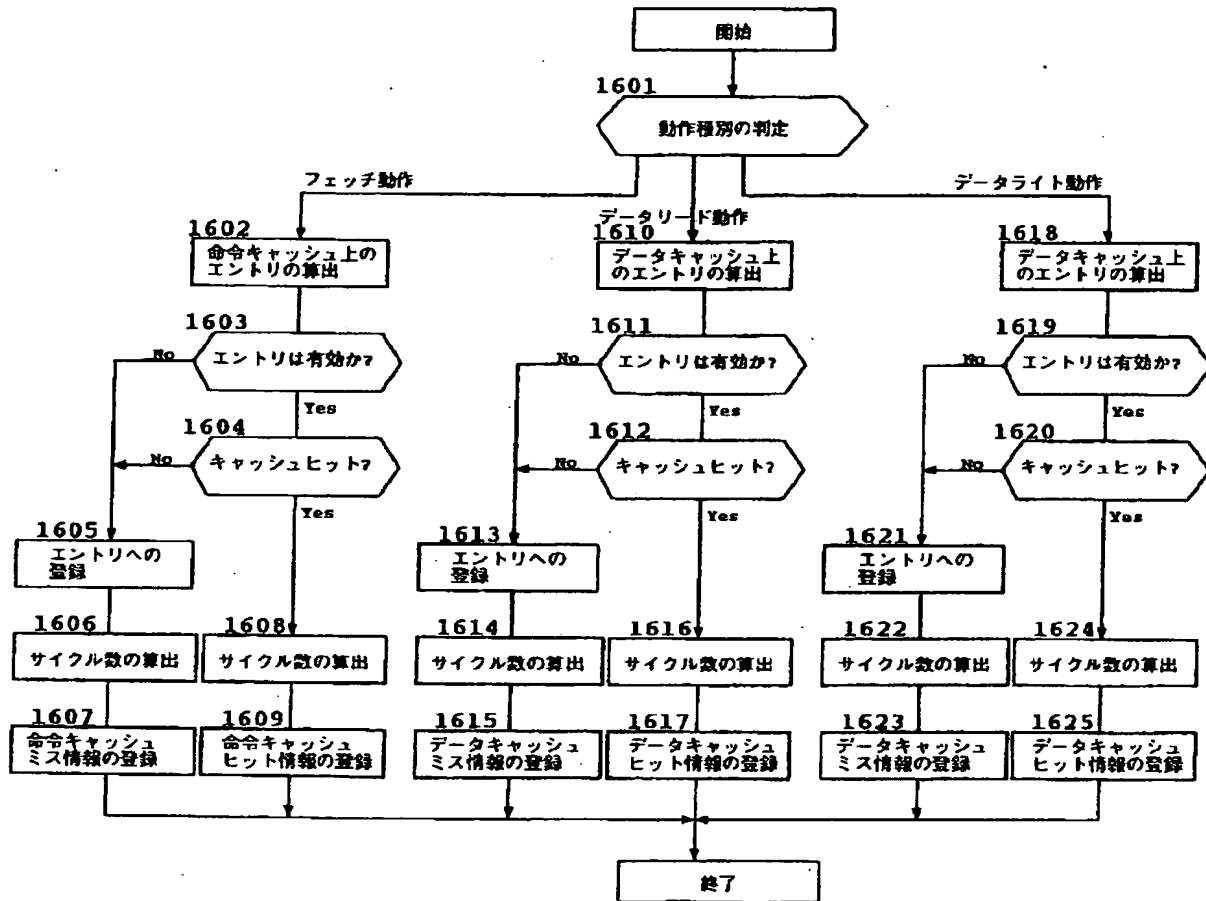
【図18】



【図 7】



【図 12】



【図 1 3】

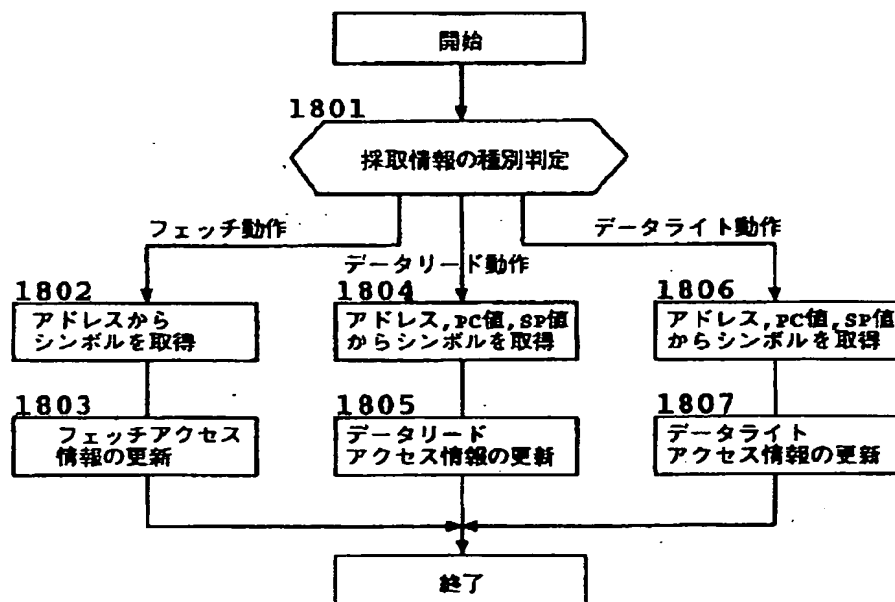
1701
キャッシュアクセス情報採取部

1702 命令キャッシュ アクセス情報管理部						
1704 フェッチアクセス情報						
entry	アドレス	Hit	データ長	シンボル	回数	割合
0x12	0x00000120	Miss	4	関数main()	1	0
0x12	0x00000124	Hit	4		1	-
0x12	0x00000128	Hit	4		1	-
0x12	0x0000012c	Hit	4		1	-

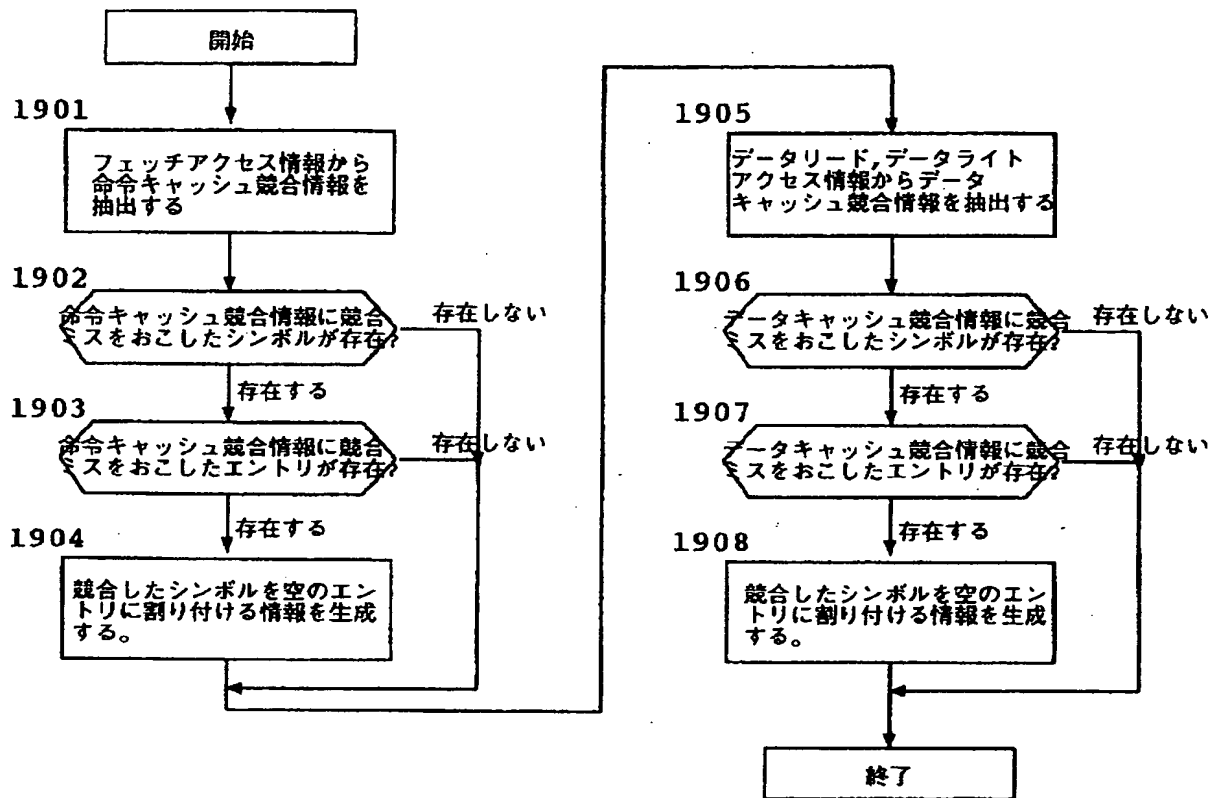
1703 データキャッシュ アクセス情報管理部						
1705 データリードアクセス情報						
entry	アドレス	Hit	データ長	シンボル	回数	割合
0x40	0x00003400	Miss	4	変数i@main	100	99
0x40	0x00003400	Hit	4	変数i@main	300	-
0x41	0x00000410	Hit	4	変数y	100	-

1706 データライトアクセス情報						
entry	アドレス	Hit	データ長	シンボル	回数	割合
0x40	0x00000400	Miss	4	変数x	101	101
0x40	0x00003400	Hit	4	変数i@main	100	-
0x40	0x00003400	Miss	4	変数i@main	1	1
0x41	0x00000410	Miss	4	変数y	1	1
0x41	0x00000410	Hit	4	変数y	100	-

【図 1 4】



【図15】



【図16】

(a)

2101 命令キャッシュ競合情報管理表

1706

entry	シンボル	競合ミスをおこしたシンボル
0x00		
0x01		
⋮		
0x10		
0x11		
0x12	関数main()	
⋮		
0x7E		
0x7F		

(b)

2102 データキャッシュ競合情報管理表

entry	シンボル	競合ミスをおこしたシンボル
0x00		
0x01		
⋮		
0x40		変数x, 変数l@main
0x41	変数y	
0x42		
⋮		
0x7E		
0x7F		

【図 1 7】

